Rlh

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/721,398	11/22/2000	Paul England	MS1-654US	2720

22801 7590 03/31/2004

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
BETIT, JACOB F	
ART UNIT	PAPER NUMBER
2175	9

DATE MAILED: 03/31/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	09/721,398	ENGLAND ET AL.
	Examiner	Art Unit
	Jacob F. Betit	2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-91 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-91 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

DOV POPOVICI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 6, 7, & 8.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Specification

1. The arrangement of the disclosed application does not conform with 37 CFR 1.77(b).

Section headings are underlined and boldfaced throughout the disclosed specification. Section headings should not be underlined and/or **boldfaced**. Appropriate corrections are required according to the guidelines provided below:

2. The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.

Arrangement of the Specification

As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.
(b) CROSS-REFERENCE TO RELATED APPLICATIONS.
(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT.
(d) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC (See 37 CFR 1.52(e)(5) and MPEP 608.05. Computer program listings (37 CFR 1.96(c)), "Sequence Listings" (37 CFR 1.821(c)), and tables having more than 50 pages of text are permitted to be submitted on compact discs.) or
REFERENCE TO A "MICROFICHE APPENDIX" (See MPEP § 608.05(a). "Microfiche Appendices" were accepted by the Office until March 1, 2001.)
(e) BACKGROUND OF THE INVENTION.
(1) Field of the Invention.
(2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.

(f) BRIEF SUMMARY OF THE INVENTION.
(g) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).
(h) DETAILED DESCRIPTION OF THE INVENTION.
(i) CLAIM OR CLAIMS (commencing on a separate sheet).
(j) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).
(k) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 9, 18, 20, 31, and 33 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. Claim 9 recites the limitation "wherein combining the different parts" in line 2. There is insufficient antecedent basis for this limitation in the claim because claim 9 is dependent on independent claim 1, which does not make the limitation "combining the different parts". For the purpose of examining, it is assumed that claim 9 was meant to be dependant on claim 8 (not claim 1).

6. Claim 18, line 2; claim 20, lines 2 and 4; claim 31, line 2; and claim 33, lines 2 and 4 recite the limitation "trusted code". There is insufficient antecedent basis for this

limitation in the claims. For the purpose of examining, it is assumed that the applicant meant --trusted core-- not "trused code".

Appropriate corrections are required.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

8. Claims 1, 6-7, 12-13, 16, 22-29, 34-39, 43-45, 57-60, 64-66, 68-74, and 79-91 are rejected under 35 U.S.C. 102(e) as being anticipated by <u>Vu et al.</u> (U.S. patent No. 6,557,104 B2.

As to claim 1, Vu et al. teaches one or more computer-readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a computer (see column 7, lines 12-30), causes the one or more processors to perform acts including:

allowing operation of the computer to begin based on untrusted code (see column 6, lines 3-21);

loading, under control of the untrusted code, a trusted core into memory (see column 6, lines 3-10);

preventing each of one or mere central processing units and each of one or more bus masters in the computer from accessing the memory (see column 4, lines 63-67);

resetting each of the one or more central processing units (see column 5, lines 27-35);

allowing one central processing unit to access the memory and execute trusted core initialization code to initialize the trusted core (see column 5, lines 33-40); and

after execution of the trusted core has been initialized, allowing any other central processing units and any bus masters in the computer to access the memory (see column 5, lines 33-40).

As to claim 6, Vu et al. teaches wherein the preventing comprises preventing each of the one or more central processing units and each of the one or more bus masters from accessing the memory in response to an initialize misted core command received from one of the one or more central processing units (see column 4, lines 63-67).

As to claim 7, Vu et al. teaches wherein the loading the misted core comprises copying different portions of the trusted core from a plurality of different sources (see column 4, lines 52-62).

As to claim 12, Vu et al. teaches wherein the loading the trusted core comprises copying at least a portion of the trusted core from a chip of the computer (see column 4, lines 52-62).

As to claim 13, Vu et al. teaches wherein the preventing comprises ignoring all requests for access to the memory from the one or more central processing units and one or more bus masters (see column 4, lines 63-67).

As to claim 16, Vu et al. teaches wherein the resetting each of the one or more central processing units comprises asserting a processor bus reset signal to each of the one or more central processing units (see column 5, lines 27-35).

As to claim 22, Vu et al. teaches wherein the plurality of instructions further cause the one or more processors to perform acts including loading microcode from the; trusted core in memory into the one central processing unit after resetting the central processing unit (see column 5, lines 33-40).

As to claim 23, Vu et al. teaches a method (see abstract) comprising:

booting, based on untrustworthy code, a computer (see column 6, lines 3-21);

loading a trusted core into memory (see column 6, lines 3-10); and

initiating secure execution of the trusted core (see column 5, lines 33-40).

As to claim 24, Vu et al. teaches further comprising:

allowing execution of the trusted core to terminate(see column 5, lines 40-43); and

re-initiating secure execution of the trusted core without re-booting the computer (see column 5, lines 24-30, where it is inherent that the "secure services routine" can be initiated whenever they are needed by calling an interrupt).

As to claim 25, Vu et al. teaches further comprising:

allowing execution of the trusted core to terminate (see column 5, lines 40-43);

loading another trusted core into memory (see column 6, lines 3-21); and

initiating secure execution of the other trusted core (see column 5, lines 33-40).

As to claim 26, Vu et al. teaches wherein the trusted core and the other trusted core are different versions of the same trusted core (see column 5, lines 33-40, where it is inherent that the other trusted core could be a different version of the same trusted core especially during the development of the trusted core).

As to claim 27, Vu et al. teaches wherein the initiating comprises initiating secure execution of the trusted core in response to an initialize trusted core command received from one of the one or more central processing units (see column 5, lines 27-34).

As to claim 28, Vu et al. teaches wherein the initiating comprises initiating secure execution of the untrusted core without requiring any additional bus transactions to be supported by processors in the computer (see column 3, line 61 thought column 4, line 10).

As to claim 29, Vu et al. teaches wherein the initiating secure execution of the trusted core comprises:

preventing each of one or more central processing units in the computer from accessing the memory (see column 4, lines 63-67);

preventing each of one or more bus .masters in the computer from accessing the memory (see column 4, lines 63-67);

resetting each of the one or more central processing units (see column 5, lines 27-35);

allowing one central processing unit to access the memory and execute a trusted core initialization process (see column 5, lines 33-40); and

after execution of the trusted core initialization process, allowing any other central processing units and any of the one or more bus masters to access the memory (see column 5, lines 33-40).

As to claim 34, Vu et al. teaches wherein the loading the trusted core comprises copying different portions of the trusted core from a plurality of different sources

including one or more of: a local mass storage device, a remote device, and a local chipset (see column 4, lines 52-62).

As to claim 35, Vu et al. teaches one or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 23 (see column 7, lines 12-30 and see rejected claim 23 above).

As to claim 36, Vu et al. teaches a method (see abstract) comprising:

allowing a computer to begin operation based on untrustworthy code (see column 6, lines 3-21);

loading, under the control of the untrustworthy code, additional code into memory (see column 6, lines 3-10); and

initiating execution of the additional code in a secure manner despite the untrustworthy code in the computer (see column 5, lines 33-40).

As to claim 37, Vu et al. teaches wherein the initiating further comprises initiating execution of the additional code in a secure manner despite both the untrustworthy code in the computer and other pre-existent state of the computer (see column 5, lines 33-40).

As to claim 38, Vu et al. teaches wherein the initiating execution of the additional code in a secure manner comprises:

preventing each of one or more central processing units in the computer from accessing the memory (see column 4, lines 63-67);

preventing each of one or more bus masters in the computer from accessing the memory (see column 4, lines 63-67);

resetting each of the one or more central processing units (see column 5, lines 27-35);

allowing one central processing unit to access the memory and execute a code initialization process (see column 5, lines 33-40); and

after execution of the code initialization process, allowing any other central processing units and any of the one or more bus masters to access the memory (see column 5, lines 33-40).

As to claim 39, Vu et al. teaches wherein the initiating comprises initiating execution of the additional code in a secure manner without requiring any additional bus transactions to be supported by a processor in the computer (see column 3, line 61 through column 4, line 10).

As to claim 43, Vu et al. teaches further comprising:

receiving, from a central processing unit, a read request corresponding to a central processing unit reset vector (see column 5, lines 27-35);

responding to the read request with instructions to cause the central processing unit to jump to a starting location of the trusted core (see column 5, lines 33-36).

As to claim 44, Vu et al. teaches wherein the loading the additional code comprises copying different portions of the additional code from a plurality of different

sources including one or more of: a local mass storage device, a remote device, and a local chipset (see column 4, lines 52-62).

As to claim 45, Vu et al. teaches one or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 36 (see column 7, lines 12-30 and see rejected claim 36 above).

As to claim 57, Vu et al. teaches an apparatus (see column 1, lines 7-10) comprising:

a processor reset portion to assert a reset signal to a processor (see column 5, lines 27-35); and

a memory protector portion to prevent any bus master from accessing memory until the processor completes execution of a misted core initialization process (see column 4, lines 63-67).

As to claim 58, Vu et al. teaches wherein the apparatus comprises a programmable logic device (see figure 5, reference number 68).

As to claim 59, Vu et al. teaches wherein the processor reset portion comprises a processor bus interface (see column 5, lines 27-35, and see figure 5).

As to claim 60, Vu et al. teaches wherein the memory protector portion comprises a control logic that ignores any request to access the memory received from any bus master (see column 4, lines 63-67).

As to claim 64, Vu et al. teaches further comprising a storage portion in which a portion of the trusted core is stored (see column 2, lines 52-62).

As to claim 65, Vu et al. teaches wherein the portion of the trusted core stored in the storage portion comprises a platform trusted core portion (see column 2, lines 52-62).

As to claim 66, Vu et al. teaches a computer comprising:

a processor; a bus master; a system memory; and a memory controller coupled to the processor, the bus master, and the system memory (see figure 5), the memory controller being configured to,

allow access to the system memory from the processor and the bus master operating based on untrustworthy code (see column 6, lines 3-21),

reset the processor to begin a trusted core initialization process (see column 5, lines 27-35), and

prevent the bus master from accessing the system memory until after the trusted core initialization process is completed (see column 4, lines 63-67).

As to claim 68, Vu et al. teaches a method comprising:

allowing execution of different trusted cores in a computer to be initiated serially without requiring the computer to be re-booted (see column 5, lines 24-48, and see column 6, lines 3-21).

As to claim 69, Vu et al. teaches wherein the allowing further comprises allowing execution of the different trusted cores to be initiated at arbitrary times (see column 5, lines 24-30, where it is inherent that the "secure services routine" can be initiated whenever they are needed by calling an interrupt).

As to claim 70, Vu et al. teaches wherein the different trusted cores are different versions of the same trusted core (see column 5, lines 33-40, where it is inherent that the other trusted core could be a different version of the same trusted core especially during the development of the trusted core).

As to claim 71, Vu et al. teaches wherein the resetting comprises asserting, on a processor bus, a RESET# signal to each of the one or more central processing units (see column 5, lines 27-35).

As to claim 72, Vu et al. teaches wherein the resetting comprises clearing a state of each of the one or more central processing units (see column 5, lines 27-35).

As to claim 73, Vu et al. teaches wherein the state of a central processing unit comprises instructions and data residing in any caches and buffers of the central processing unit (see column 5, lines 27-35).

As to claim 74, Vu et al. teaches wherein the state of a central processing unit comprises instructions and data residing in any registers of the central processing unit (see column 5, lines 27-35).

As to claim 79, Vu et al. teaches wherein the reset signal clears a state of the processor (see column 5, lines 27-35).

As to claim 80, Vu et al. teaches wherein the state of the processor includes instructions and data residing in any caches or buffers of the processor (see column 5, lines 27-35).

As to claim 81, Vu et al. teaches wherein the processor reset portion is to assert the reset signal on a processor bus (see column 5, lines 27-35).

As to claim 82, Vu et al. teaches wherein the reset signal comprises RESET# (see column 5, lines 27-35).

As to claim 83, Vu et al. teaches wherein the memory controller is further configured to reset the processor by clearing a state of the processor (see column 5, lines 27-35).

As to claim 84, Vu et al. teaches wherein the state of the processor includes instructions and data residing in any caches and buffers of the processor (see column 5, lines 27-35).

As to claim 85, Vu et al. teaches wherein the state of the processor includes instructions and data residing in any registers of the processor (see column 5, lines 27-35).

As to claim 86, Vu et al. teaches wherein the memory controller is further configured to reset the; processor by asserting, on a processor bus, a reset signal to the processor (see column 5, lines 27-35).

As to claim 87, Vu et al. teaches wherein the memory controller is further configured to reset the processor by asserting a RESET# signal to the processor (see column 5, lines 27-35).

As to claim 88, Vu et al. teaches a method (see abstract) comprising:

allowing operation of a computer to begin based on untrusted code (see column 6, lines 3-21);

loading, under control of the untrusted code, a trusted core into memory of the computer (see column 6, lines 3-10);

preventing each of one or more central processing units and each of one or more bus masters in the computer from accessing the memory (see column 4, lines 63-37);

clearing a state of each of the one or more central processing units (see column 5, lines 27-35);

allowing one central processing unit to access the memory and execute trusted core initialization code to initialize the trusted core (see column 5, lines 33-40); and

after execution of the trusted core has been initialized, allowing any other central processing units and any bus masters in the computer to access the memory (see column 5, lines 33-40).

As to claim 89, Vu et al. teaches wherein the preventing comprises preventing each of the one or more central processing units and each of the one or more bus masters from accessing the memory in response to an initialize trusted core command received from one of the one or more central processing units (see column 4, lines 63-67).

As to claim 90, Vu et al. teaches wherein the state of a central processing unit comprises instructions and data residing in any caches and buffers of the central processing unit (see column 5, lines 27-35).

As to claim 91, Vu et al. teaches wherein the state of a central processing unit comprises instructions and data residing in any registers of the central processing unit (see column 5, lines 27-35).

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 46, 48-52, and 75-78 are rejected under 35 U.S.C. 102(b) as being anticipated by Mattison (U.S. patent No. 5,778,070).

As to claim 46, Mattison teaches a memory controller (see figure 2, reference number 104) comprising:

a first interface to allow communication with a processor (see figure 2, reference numbers 202 and 204);

a second interface to allow communication with a system memory (see figure 2, reference numbers 206 and 208); and

a controller, coupled to the first interface and the second interface, to reset a processor and to allow the processor to execute a code initialization process while preventing any other processors from accessing the system memory (see column 8, lines 29-38).

As to claim 48, Mattison teaches wherein the first interface comprises a processor bus interface (see figure 2, reference numbers 202 and 204).

As to claim 49, Mattison teaches wherein the memory controller operates without requiring the processor bias interface to support any additional commands on the processor bus (see column 8, lines 29-38).

As to claim 50, Mattison teaches wherein the system memory comprises a dynamic random access memory (see figure 2, reference number 106, where it is inherent that most modern computer systems use "dynamic random access memory" for system memory.

As to claim 51, Mattison teaches wherein the controller is further to allow the processor to execute the code initialization process while preventing any bus masters from accessing the system memory (see column 8, lines 39-60).

As to claim 52, Mattison teaches a memory controller as recited in claim 46, wherein the controller is further to:

reset any other processor coupled to the memory controller prior to allowing the processor to execute the code initialization process (see column 8, lines 27-38);

prevent any other processor and any bus master coupled to the memory controller from accessing the system memory until the one process executes the code initialization process (see column 8, lines 39-60); and

after execution of the code initialization process, allow any other central processing units coupled to the memory controller and any bus masters coupled to the memory controller to access the memory (see column 4, lines 8-14).

As to claim 75, Mattison teaches wherein the controller is to reset the processor by clearing a stage of the processor (see column 8, lines 29-38).

As to claim 76, Mattison teaches wherein the clearing the state of the processor comprises clearing all instructions and data from any caches or buffers of the processor (see column 8, lines 29-38).

As to claim 77, Mattison teaches wherein the controller is to reset the processor by asserting, on a processor bus, a reset signal to the processor (see column 8, lines 29-38).

As to claim 78, Mattison teaches wherein the reset signal comprises RESET# (see column 8, lines 29-38).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 2-4, 17-21, 30-33, 40-42, and 62-63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vu et al. (U.S. patent No. 6,557,104 B2) in view of Virajpet et al. (U.S. patent No. 6,480,948 B1).

As to claim 2, Vu et al. does not teach wherein the one or more processors comprise one or more controllers of one or more memory controllers.

Virajpet et al. teaches wherein the one or more processors comprise one or more controllers of one or more memory controllers (see column 5, lines 17-42).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include wherein the one or more processors comprise one or more controllers of one or more memory controllers.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Virajpet et al. because wherein the one or more processors comprise one or more controllers of one or more memory controllers would allow the processor to write values in the memory controller to change the memory map (see Virajpet et al., column 5, lines 17-42).

As to claim 3, Vu et al. as modified, teaches wherein the one or more memory controllers are distributed among the one or more central processing units (see Virajpet et al., figure 1, reference numbers 10 and 12).

As to claim 4, Vu et al. as modified, teaches wherein the plurality of instructions comprise microcode to be executed by the one or more memory controllers (see Virajpet et al., column 5, lines 17-42).

As to claim 17, Vu et al. teaches wherein the plurality of instructions further cause the one or more processors to perform acts including:

receiving a read request corresponding to the central processing unit reset vector from the one central processing unit (see column 5, lines 27-35);

returning, in response to the read request, the initialization vector to the one central processing unit (see column 5, lines 33-36); and

allowing the one central processing unit to access the memory beginning with the initialization vector (see column 5, lines 35-40).

Vu et al. does not teach mapping a central processing unit reset vector to an initialization vector.

Virajpet et al. teaches mapping a central processing unit reset vector to an initialization vector (see column 3, lines 18-28).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include mapping a central processing unit reset vector to an initialization vector.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Virajpet et al. because mapping a central processing unit reset vector to an initialization vector would

make access to interrupt code faster by accessing code in SRAM instead of ROM (see Virajpet et al., column 3, lines 18-28).

As to claim 18, Vu et al. as modified, teaches wherein the initialization vector is an address within the trusted core in the memory (see Vu et al., column 5, lines 35-40).

As to claim 19, Vu et al. as modified, teaches wherein the plurality of instructions further cause the one or more processors to perform acts including:

re-mapping the central processing unit reset vector to an additional central processing unit start vector after returning the initialization vector to the one central processing unit (see Virajpet et al., column 5, lines 37-42); and

returning, in response to any other read request corresponding to the central processing unit reset vector from another central processing unit, the additional central processing unit start vector (see Vu et al., column 5, lines 33-36).

As to claim 20, Vu et al. as modified, teaches wherein the initialization vector is an address within the trusted core in the memory and wherein the additional central processing unit start vector and the initialization vector are different addresses within the trusted core in the memory (see Vu et al., column 5, lines 33-40).

As to claim 21, Vu et al. as modified, wherein both the initialization vector and the additional central processing unit start vector are obtained from the trusted core (see Virajpet et al., column 5, lines 37-42).

As to claim 30, Vu et al. teaches further comprising:

receiving a read request corresponding to the central processing unit reset vector from the one central processing unit (see column 5, lines 27-35);

returning, in response to the read request, the initialization vector to the one central processing unit (see column 5, lines 33-36); and

allowing the one central processing unit to access the memory beginning with the initialization vector (see column 5, lines 35-40).

Vu et al. does not teach mapping a central processing unit reset vector to an initialization vector.

Virajpet et al. teaches mapping a central processing unit reset vector to an initialization vector (see column 3, lines 18-28).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include mapping a central processing unit reset vector to an initialization vector.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Virajpet et al. because mapping a central processing unit reset vector to an initialization vector would make access to interrupt code faster by accessing code in SRAM instead of ROM (see Virajpet et al., column 3, lines 18-28).

As to claim 31, Vu et al. as modified, teaches wherein the initialization vector is an address within the trusted core in, the memory (see Vu et al., column 5, lines 35-40).

As to claim 32, Vu et al. as modified, teaches further comprising:

re-mapping the central processing unit reset vector to an additional central processing unit start vector after returning the initialization vector to the one central processing unit (see Virajpet et al., column 5, lines 37-42); and

returning, in response to any other read request corresponding to the central processing unit reset vector from another central processing unit, the additional central processing unit start vector (see Vu et al., column 5, lines 33-36).

As to claim 33, Vu et al. as modified, teaches wherein the initialization vector is an address within the trusted core in the memory and wherein the additional central processing unit start vector and the initialization vector are different addresses within the trusted core in the memory (see Virajpet et al., column 5, lines 37-42).

As to claim 40, Vu et al. teaches further comprising:

receiving a read request corresponding to the central processing unit reset vector from the one central processing unit (see column 5, lines 27-35);

returning, in response to the read request, the initialization vector to the one central processing unit (see column 5, lines 33-36); and

allowing the one central processing unit to access the memory beginning with the initialization vector (see column 5, lines 35-40).

Vu et al. does not teach mapping a central processing unit reset vector to an initialization vector.

Virajpet et al. teaches mapping a central processing unit reset vector to an initialization vector (see column 3, lines 18-28).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include mapping a central processing unit reset vector to an initialization vector.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Virajpet et al. because mapping a central processing unit reset vector to an initialization vector would make access to interrupt code faster by accessing code in SRAM instead of ROM (see Virajpet et al., column 3, lines 18-28).

As to claim 41, Vu et al. as modified, teaches further comprising:

re-mapping the central processing unit reset vector to an additional central processing unit start vector after returning the initialization vector to the one central processing unit (see Virajpet et al., column 5, lines 37-42); and

returning, in response to any other read request corresponding to the central processing unit reset vector from another central processing unit, the additional central processing unit start vector (see Vu et al., column 5, lines 33-36).

As to claim 42, Vu et al. does not teach further comprising:

remapping the trusted core to appear at an address where a central processing unit starts executing after being reset.

Virajpet et al. teaches further comprising: remapping the trusted core to appear at an address where a central processing unit starts executing after being reset (see column 3, lines 1-28).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include further comprising: remapping the trusted core to appear at an address where a central processing unit starts executing after being reset.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Virajpet et al. because further comprising: remapping the trusted core to appear at an address where a central processing unit starts executing after being reset would make access to interrupt code faster by accessing code in SRAM instead of ROM (see Virajpet et al., column 3, lines 18-28).

As to claim 62, Vu et al. as modified, teaches further comprising a controller, coupled to the memory protector portion, to:

receive a read request corresponding; to the processor reset vector from the processor (see column 5, lines 27-35);

return, in response to the read request, the initialization vector to the processor (see column 5, lines 33-36); and

allow the processor to access the memory beginning with the initialization vector (see column 5, lines 35-40).

Vu et al. does not teach map a processor reset vector to an initialization vector.

Virajpet et al. teaches map a processor reset vector to an initialization vector (see column 3, lines 18-28).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include map a processor reset vector to an initialization vector;

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Virajpet et al. because map a processor reset vector to an initialization vector would make access to interrupt code faster by accessing code in SRAM instead of ROM (see Virajpet et al., column 3, lines 18-28).

As to claim 63, Vu et al. as modified, teaches wherein the controller is further to:

re-map the processor reset vector to an additional processor start vector after returning the initialization vector to the processor (see Virajpet et al., column 5, lines 37-42); and

return, in response to another read request corresponding to the processor reset vector from another processor, the additional processor start vector (see Vu et al., column 5, lines 33-36).

13. Claims 5 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vu et al. (U.S. patent No. 6,557,104 B2) in view of Frank, Jr. et al. (U.S. patent No. 6,546,489 B1).

As to claim 5, Vu et al. does not teach wherein the untrusted code includes code from a basic input output system (BIOS) and code from a plurality of option read only memories (ROMs).

Frank, Jr. et al. teaches wherein the untrusted code includes code from a basic input output system (BIOS) and code from a plurality of option read only memories (ROMs) (see column 4, line 66 through column 5, line 6).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include wherein the untrusted code includes code from a basic input output system (BIOS) and code from a plurality of option read only memories (ROMs).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Frank, Jr. et al. because wherein the untrusted code includes code from a basic input output system (BIOS) and code from a plurality of option read only memories (ROMs) would realize more of the types of memory that are subject to contamination (see Frank, Jr. et al., column 5, lines 3-6).

As to claim 10, Vu et al. does not teach wherein the loading the trusted core comprises copying at least a portion of the trusted core from a local mass storage device into the memory.

Frank, Jr. et al. teaches wherein the loading the trusted core comprises copying at least a portion of the trusted core from a local mass storage device into the memory (see column 5, lines 31-45).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include wherein the loading the trusted core comprises copying at least a portion of the trusted core from a local mass storage device into the memory.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Frank, Jr. et al. because wherein the loading the trusted core comprises copying at least a portion of the trusted core from a local mass storage device into the memory would allow the host computer to be activated with a memory image source whose source is impervious to virus or inadvertent corruption (see Frank, Jr. et al., abstract).

14. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vu et al. (U.S. patent No. 6,557,104 B2) in view of Faber et al. patent No. 6,477,252 B1).

As to claim 8, Vu et al. does not teach wherein the loading the trusted core comprises copying different parts of the trusted core from one or more sources and combining the different parts to assemble the trusted core.

Faber et al. teaches wherein the loading the trusted core comprises copying different parts of the trusted core from one or more sources and combining the different parts to assemble the trusted core (see figure 3, step 318).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include wherein the

loading the trusted core comprises copying different parts of the trusted core from one or more sources and combining the different parts to assemble the trusted core.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Faber et al. because wherein the loading the trusted core comprises copying different parts of the trusted core from one or more sources and combining the different parts to assemble the trusted core would protect the content of the data stream (see Faber et al., column 1, lines 9-13).

As to claim 9, Vu et al. as modified, teaches wherein combining the different parts comprises exclusive-ORing bits of the different parts (see Faber et al., figure 3, step 318).

15. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Vu et al. (U.S. patent No. 6,557,104 B2) in view of Cox et al. patent No. 5,349,643).

As to claim 11, Vu et al. does not teach wherein the loading the trusted core comprises copying at least a portion of the trusted core from a remote device into the memory.

Cox et al. teaches wherein the loading the trusted core comprises copying at least a portion of the trusted core from a remote device into the memory (see abstract).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include wherein the

loading the trusted core comprises copying at least a portion of the trusted core from a remote device into the memory.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Cox et al. because wherein the loading the trusted core comprises copying at least a portion of the trusted core from a remote device into the memory would allow secure boot for a diskless workstation (see Cox et al., column 1, lines 7-10).

16. Claims 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vu et al. (U.S. patent No. 6,557,104 B2) in view of Collins et al. patent No. 6,378,072 B1).

As to claim 14, Vu et al. does not teach wherein the plurality of instructions further cause the one or more processors to perform acts including:

extracting a cryptographic measure of the trusted core in the memory; and

storing the extracted cryptographic measure.

Collins et al. teaches wherein the plurality of instructions further cause the one or more processors to perform acts including: extracting a cryptographic measure of the trusted core in the memory (see column 9, lines 12-42); and storing the extracted cryptographic measure (see column 9, lines 54-60, where it is inherent that the checksum developed from the original form of the program file would be stored locally to be compared with the one generated by the cryptographic processor).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include wherein the

plurality of instructions further cause the one or more processors to perform acts including: extracting a cryptographic measure of the trusted core in the memory; and storing the extracted cryptographic measure.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Collins et al. because wherein the plurality of instructions further cause the one or more processors to perform acts including: extracting a cryptographic measure of the trusted core in the memory; and storing the extracted cryptographic measure would allow for a check to make sure the program file is authentic (see Collins et al., column 9, lines 12-20).

As to claim 15, Vu et al. as modified, teaches wherein the plurality of instructions further cause the one or more processors to perform acts including:

resetting a cryptographic processor (see Collins et al., column 8, lines 12-28);

requesting the cryptographic processor to extract the cryptographic measure (see Collins et al., column 9, lines 12-42); and

receiving the extracted cryptographic measure from the cryptographic processor (see Collins et al., column 9, lines 32-36 and see lines 54-60).

17. Claim 47 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mattison (U.S. patent No. 6,615,355 B2) in view of 486 Microprocessors, SSV Software Systems PC/104 Products, http://www.ssv-embedded.de/ssv/pc104/p71.htm, March 13, 1998 (hereinafter referred to as 486 Microprocessors).

As to claim 47, Mattison does not teach wherein the memory controller is included in a processor.

486 Microprocessors teaches wherein the memory controller is included in a processor (see page 2, 486 Cache Unit).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Mattison to include wherein the memory controller is included in a processor.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Mattison by the teachings of 486 Microprocessors because wherein the memory controller is included in a processor would allow control of the onboard cache memory on the processor.

18. Claims 53-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattison (U.S. patent No. 6,615,355 B2) in view of Virajpet et al. (U.S. patent No. 6,480,948 B1) and further in view of Vu et al. (U.S. patent No. 6,557,104).

As to claim 53, Mattison does not teach wherein the controller is further to:

map a processor reset vector to an initialization vector;

Virajpet et al. teaches map a processor reset vector to an initialization vector (see column 3, lines 18-28).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Mattison to include map a processor reset vector to an initialization vector.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Mattison by the teachings of Virajpet et al. because map a processor reset vector to an initialization vector would make access to interrupt code faster by accessing code in SRAM instead of ROM (see Virajpet et al., column 3, lines 18-28).

Mattison as modified, still does not teach receive a read request corresponding to the processor reset vector from the processor; return, in response to the read request, the initialization vector to the processor; and allow the processor to access the memory beginning with the initialization vector.

Vu et al. teaches receive a read request corresponding to the processor reset vector from the processor (see column 5, lines 27-35); return, in response to the read request, the initialization vector to the processor (see column 5, lines 33-36); and allow the processor to access the memory beginning with the initialization vector (see column 5, lines 35-40).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Mattison as modified, to include receive a read request corresponding to the processor reset vector from the processor; return, in response to the read request, the initialization vector to the processor; and allow the processor to access the memory beginning with the initialization vector.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Mattison as modified, by the teachings of Vu et al. because receive a read request corresponding to the processor reset vector from the processor; return, in response to the read request, the initialization vector to the

processor; and allow the processor to access the memory beginning with the initialization vector would all the processor to access a secure processing mode (see abstract).

As to claim 54, Mattison as modified, teaches wherein the initialization vector is an address within the code initialization process (see Vu et al., column 5, lines 35-40).

As to claim 55, Mattison as modified, teaches wherein the controller is further to:

re-map the processor reset vector to an additional processor start vector after returning the initialization vector to the processor (see Vu et al., column 5, lines 33-36); and

return, in response to any other read request corresponding to the processor reset vector from another processor, the additional processor start vector (see Vu et al., column 5, lines 33-36).

As to claim 56, Mattison as modified, teaches wherein the initialization vector is an address within the code initialization process and wherein the additional processor start vector and the initialization vector are different addresses within the code initialization process (see column Vu et al., 5, lines 35-40).

19. Claims 61 and 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vu et al. (U.S. patent No. 6,557,104 B2) in view of Stumpf et al. (U.S. patent No. 5,175,829).

As to claim 61, Vu et al. does not teach further comprising a controller, coupled to the memory protector portion, to prevent another processor from accessing memory until the processor completes execution of the trusted core initialization process.

Stumpf et al. teaches further comprising a controller, coupled to the memory protector portion, to prevent another processor from accessing memory until the processor completes execution of the trusted core initialization process (see abstract).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include further comprising a controller, coupled to the memory protector portion, to prevent another processor from accessing memory until the processor completes execution of the trusted core initialization process.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Stumpf et al. because further comprising a controller, coupled to the memory protector portion, to prevent another processor from accessing memory until the processor completes execution of the trusted core initialization process would stop any other processors from accessing that section of memory during atomic operations (see Stumpf et al., abstract).

As to claim 67, Vu et al. does not teach further comprising a plurality of additional processors and preventing the plurality of additional processors from accessing the system memory until after the trusted core initialization process is completed.

Stumpf et al. teaches further comprising a plurality of additional processors and preventing the plurality of additional processors from accessing the system memory until after the trusted core initialization process is completed (see abstract).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. to include further comprising a plurality of additional processors and preventing the plurality of additional processors from accessing the system memory until after the trusted core initialization process is completed.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Vu et al. by the teachings of Stumpf et al. because further comprising a plurality of additional processors and preventing the plurality of additional processors from accessing the system memory until after the trusted core initialization process is completed would stop any other processors from accessing that section of memory during atomic operations (see Stumpf et al., abstract).

Conclusion

20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob F. Betit whose telephone number is (703) 305-3735. The examiner can normally be reached on Monday through Friday 9 am to 5 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dov Popovici can be reached on (703) 305-3830. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jfb
19 March 2004

DOV POPOVICI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100